Xen Install. Mini how-to.

December 8, 2005

1 Introduction

2 Getting the latest Xen sources

Xen uses Mercurial(hg) repository. Here is an original how-to from Xen team describing how to use it is here:
http://www.xensource.com/xen/documentation/hg-cheatsheet.txt.
However, actually you need only three steps to download latest version of Xen.

Initialize repository:

```
hg init
```

Configure the repository root (parent) by putting the following into .hg/hgrc: For Xen 3.0 (unstable):

```
[paths]
default = http://xenbits.xensource.com/xen-unstable.hg
```

or for Xen 2.0:

```
[paths]
default = http://xenbits.xensource.com/xen-2.0.hg
```

Pull the most recent changes:

```
hg pull -u
```

2.1 Compiling Xen

Root directory of the Xen tree contains README file describing how to compile Xen. However, most of the time you just have to run:

```
make world
make install
```

This will put Xen related configuration files into /etc/xen and install Xen dom0 and domU kernels into /boot directories. Further information about build process is available in the README file in root directory of Xen source tree.

Note, that the README file recommends to build Xen with the following command: make KERNELS=linux-2.6-xen world. README file says that it will build Xen kernel capable to run in both dom0 and domU instead of two separate kernels. Unfortunately, I believe, it contains an error, and in contrast to README omitting KERNELS=linux-2.6-xen argument to make install you should invoke both commands with this argument otherwise it will not install your kernel.

```
make KERNELS=linux-2.6-xen world
make KERNELS=linux-2.6-xen install
```

Currently kernel built in this way hangs on boot. So far I'm unable to understand what's wrong. Therefore we have to start from the two kernel configuration. Another argument for doing that is optimization. For us it's probably better to have separate kernels optimized for domU and dom0.

Note, that you built one kernel, you have to adjust dom0 kernel name in grub.conf accordingly.

Note, do not try to build such large project in NFS, build will most probably fail.

2.2 Configuring GRUB

Xen uses GRUB loader for booting. Thus you have at first install GRUB (I can write about that later), and then edit GRUB configuration file (/boot/grub/grub.conf) to look like:

```
default 0
timeout 5

# We can control GRUB via COM1.
# Disable by default as it hangs on systems without a serial port
serial –unit=0 –speed=115200 –word=8 –parity=no –stop=1
terminal –timeout=5 serial console

title Xen 3.0 / XenoLinux 2.6
kernel /boot/xen-3.gz dom0_mem=131072 com1=115200,8n1 noirqbalance
module /boot/vmlinuz-2.6-xen0 root=/dev/sda2 ro console=ttyS0
module /boot/initrd-2.6.12-1.1390_FC4-emulab-1.img
```

Parameter noirqbalance above is a hack to prevent Dell xxxx box from rebooting (see [1, 2] for details). Note also, that I used initrd-2.6.12-1.1390_FC4-emulab-1.img from Emulab's installation of Fedora Core, however you can generate the new initrd ram image by invoking the following commands

```
depmod 2.6.12.6-xen mkinitrd -v -f –with=aacraid –with=sd_mod –with=scsi_mod
initrd-2.6.12.6-xen.img 2.6.12.6-xen
```

You can find some additional information in the README file from the root of Xen source directory.

3 Starting VM Environment

After rebooting into Xen you have to start xend - Xen control daemon by invoking:

```
xend start
```

You can configure xend to start certain domains automatically (look Section 3.3 Starting and Stopping Domains Automatically in the Xen User's Manual
http://www.cl.cam.ac.uk/Research/SRG/netos/xen/readmes/user/user.html for how to do that.

After starting xend you can configure and start VMs (Xen domUs).

3.1 Default gateway bug

Currently, xend script has the following error: on startup it destroys default route to 155.98.36.1 (control-router.emulab.net) what efficiently makes Emulab node unreachable (as well as brakes your connection). So far, I invoke it as follows:

```
xend start; route add default gw 155.98.36.1
```

I hoped, that Xen team will fix this bug, and haven't payed attention to it, but now it's time to understand what's wrong.

4 Starting VMs

In order to run the simplest VM you have to create a disk image to boot from. Note, that you can boot VM from your physical partition in read only or even in write mode however it doesn't make much sense for us. Therefore I will consider only situations when you plan to start Xen VMs from separate disk images.

4.1 Creating Xen bootable image

The following commands were taken from the following how-to (unfortunately it was modified not in a good way recently):
http://wiki.xensource.com/xenwiki/InstallGuestImage

4.1.1 Creating slice image

Create image file (here I assume 3GB file)

```
dd if=/dev/zero of=hd.img bs=1M count=1 seek=3072
```

Associate a loopback device with the file

```
losetup /dev/loop1 hd.img
```

Install filesystem

```
mkfs.ext3 /dev/loop1
```

Mount host operating system and new disk

```
mkdir -p /mnt/host
mkdir -p /mnt/guest
mount /dev/sda2 /mnt/host
mount /dev/loop1 /mnt/guest
```

Copy files and create necessary directories

```
cp -ax /mnt/host/{root,dev,var,etc,usr,bin,sbin,lib} /mnt/guest
mkdir /mnt/guest/{proc,sys,home,tmp}
```

Unmount host

```
umount /mnt/host
```

Edit /etc/fstab to look something like

```
/dev/sda1     /              ext3   defaults                        1 1
/dev/devpts   /dev/pts       devpts gid=5,mode=620                  0 0
/dev/shm      /dev/shm       tmpfs  defaults                        0 0
/dev/proc     /proc          proc   defaults                        0 0
/dev/sys      /sys           sysfs  defaults                        0 0
/dev/sda3     swap           swap   defaults                        0 0
/dev/fd0      /media/floppy  auto   pamconsole,exec,noauto,managed 0 0
/dev/hdc      /media/cdrom   auto   pamconsole,exec,noauto,managed 0 0
```

Unmount image file

```
umount /mnt/guest
losetup -d /dev/loop1
```

Now you are able to pass this file as a partition image to Xen VM mounting it via a loopback device

```
losetup /dev/loop1 hd.img
```

4.1.2 Creating disk image

4.2 Using LVM

As in the previous sections we will create LVM device on the file. To do that create large file for LVM device

```
dd if=/dev/zero of=lvm.img bs=1M count=1 seek=3072
```

Mount it via a loopback device

```
losetup /dev/loop1 lvm.img
```

Initialize partition to support LVM volumes

```
pvcreate /dev/loop1
```

Create an LVM volume group called vg

```
vgcreate vg /dev/loop1
```

Create a logical volume called myvmdisk1 of size 2500MB

```
lvcreate -L2500M -n myvmdisk1 vg
```

Now you should now see /dev/vg/myvmdisk1 device. Make a filesystem, mount it and populate it, e.g.

```
mkfs -t ext3 /dev/vg/myvmdisk1
```

Copy root partition on LVM volume and create necessary directories on it

```
mount /dev/vg/myvmdisk1 /mnt/guest
mount /dev/sda2 /mnt/host
cp -ax /mnt/host/{root,dev,var,etc,usr,bin,sbin,lib} /mnt/guest
mkdir /mnt/guest/{proc,sys,home,tmp}
```

Unmount everything

```
umount /mnt/guest
umount /mnt/host
```

Don't forget to edit fstab file and do other changes to guest VM image.

Create a couple of 100MB copy-on-write clones

```
lvcreate -s -L100M -n myclonedisk1 /dev/vg/myvmdisk1
lvcreate -s -L100M -n myclonedisk2 /dev/vg/myvmdisk1
```

Now you are able to export LVM COW disk to VM as

disk = ['phy:/dev/vg/myclonedisk1,sda1,w']

Note, that in this example original LVM image was pretty small (only 3GB), thus we created very small (only 100MB) COW shadow images. We have to think about how large should be the shadows in Emulab.

Note, that you can enlarge the shadow by invoking

lvextend +100M /dev/vg/myclonedisk1

However, all shadows and parent images should fit into original LVM 3GB image

4.3 Configuration files

In order to start Xen VM you have to create configuration file for it. The simple configuration file can look like:

```
kernel ="/boot/vmlinuz-2.6.12-xenU"
memory = 128
name = "VM1"
cpu=1
disk = ['phy:/dev/loop1,sda1,w']
vif = [ 'bridge=xen-br-routeA', 'bridge=xen-br-routeAB' ]
root = "/dev/sda1 ro"
ramdisk = "/tmp/sda4/images/initrd-fc3.img"
extra = "ro selinux=0 3"
```

Here VM is configured to have 128MB of memory and emulate single CPU machine. The physical device /dev/loop1 will be exported to VM as sda1 device. Two virtual interfaces: bridge=xen-br-routeA, bridge=xen-br-routeAB will be also exported to VM, usually as well as in the example above, interfaces exported to VM are Linux bridges. Note, that VM name should be unique.

initrd is passed to Fedora Core, which requires it to boot. Other quest OSes don't need it (see section 2.2 for how to create initrd ram image).

Note that you can use vif variable to specify MAC address for the VM's NIC

vif = ['mac=00:16:3E:F6:BB:B3']

A couple of example configuration files installed with Xen VMM can be found in /etc/xen/.

Note, Xen documentation says you can export a filesystem to domU directly via a file containing this filesystem. Thus if you have a disk image (full disk with partition table, partitions, boot record etc.) you can pass it as:

disk = ['file:/tmp/sda4/images/router_A.img,sda,w']

Or if you have an image containing only one partition

disk = ['file:/tmp/sda4/images/sda1.img,sda1,w']

Unfortunately, I wasn't able to figure out the way how to do that correctly, both disk and partition images either hang domU on boot, as in case of disk image (domU is unable to find root on the disk) or report that virtual block device (vbd) cannot be connected as in case of single slice image. For more information on that issue look Section 6.2 Using File-backed VBDs in Xen User's manual: http://www.cl.cam.ac.uk/Research/SRG/netos/xen/readmes/user/user.html

4.4 Starting VM

In order to start VM you have to invoke the following command

xm create <configuration file>

In order to access console of the started VM invoke

xm console <VM id>

Where <VM id> can be learned by running

xm list

Alternatively you can pass -c option to xm upon VM creation and connect to VM's console immediately after start.

4.4.1 /lib/tls bug

Upon the boot both dom0 and domU complain that tls emulation is slow and recommend to disable tls by renaming system library, i.e.

mv /lib/tls /lib/tls.disabled

In my case, this step hasn't made any effect, however I remember that it worked for Xen 2.0. That's strange and I have to spend some time figuring what's going on.

4.5 Setting up the bridge

Although you can create and configure bridge by yourself (look man brctl for information) Xen provides scripts for basic bridge creation and configuration. Thus, the following command creates the xen-br-routeA bridge:

/etc/xen/scripts/network-bridge start bridge=xen-br1 netdev=eth5 antispoof=no

Later you can export this bridge to Xen VM by mentioning it in configuration file (see section 4.3).

5 Setting up the virtual LAN

5.1 Recompiling your kernel to use VLAN

In order to use VLANs you have to recompile your dom0 kernel with 802.1q enabled. Like for any other Linux kernel option, there are two ways to do that: either through the GUI configuration environment or by editing .config file. In first case, you have to run

```
make linux-2.6-xen0-config CONFIGMODE=menuconfig
```

And following the path "Device drivers->Networking support->Networking options->802.1Q-> VLAN Support" enable 802.1Q support.

Alternatively you can directly edit Linux kernel configuration file .config in linux-2.6.12-xen0 directory and set

```
CONFIG_VLAN_8021Q=m
```

what means compile 802.1Q as a module.

In both cases, you have to recompile and reinstall your kernel by invoking

```
make linux-2.6-xen-build
make linux-2.6-xen-install
```

5.2 Configuring the virtual LAN

The following four commands create a bridge attached to the VLAN with id 573:

```
vconfig add eth2 573
ifconfig eth2.573 up
/etc/xen/scripts/network-bridge start bridge=xen-br1 netdev=eth2.573 antispoof=no
brctl addif xen-br1 eth2.573
```

First command creates VLAN device on the eth2 interface. VLAN device name will be eth2.573. Next command brings it up, after which brctl bridge is created on this device the help of Xen network-bridge script. Note, what is strange, Xen network-bridge script doesn't attach device eth2.573 to the new bridge, therefore fourth command attaches the bridge explicitly.

5.3 Checksum offloading bug

In order to improve network I/O performance Xen drivers allow VMs to use checksum offloading assuming that when packet is be transmitted between VMs on the same physical host actual checksum need not to be calculated. In case when packet lives physical node, Xen driver passes packet to actual hardware which calculates checksum.

Unfortunately, this fails for packets transmitted over VLAN. In this case checksum will not be calculated and receiver on the other side of the VLAN will drop the packet.

In order to prevent that I explicitly disable checksum offloading inside VM by invoking:

ethtool -K eth0 tx off

However, the better solution will be to either recompile Linux kernel or configure the system to not offload the checksum.

6 Traffic shaping

6.1 Recompiling your kernel

In order to use traffic shaping you also have to change configuration and recompile dom0 Linux kernel (see section 5.1 for details).

Do do that modify "Device drivers->Networking support->Networking options-> QoS and/or fair queueing" and enable bunch of related options, unfortunately, I don't have exact knowledge of which option means what and currently enable most of them. I have to spend some time on this issue.

6.2 tc

My experience in traffic shaping with tc is based on the following how-to
http://linux-net.osdl.org/index.php/Netem.

I still don't know whether tc evolved enough to support ingress shaping, if not we have to adopt Kirk's solution in dom0.

I assume that the good way to shape traffic between VMs is to apply tc rules to the virtual interfaces exported to VMs. After VM boot they always take form of vif<vm id>.<interface number>

7 Related resources

Xen at Clarkson
http://xen.cosi.clarkson.edu/

Xen User's Manual
http://www.cl.cam.ac.uk/Research/SRG/netos/xen/readmes/user/user.html

Mercurial(hg) Cheatsheet for Xen
http://www.xensource.com/xen/documentation/hg-cheatsheet.txt

Install Guest Image
http://wiki.xensource.com/xenwiki/InstallGuestImage

8 References

[1] xen-devel mailing list. (20 Jul 2005) [Xen-devel] Xen / Dell 2850 PERC 4e/Di lock up
http://lists.xensource.com/archives/html/xen-devel/2005-07/msg00659.html

[2] Xen Bugzilla database (fixed 2005-08-29):
http://bugzilla.xensource.com/bugzilla/show_bug.cgi?id=76